Noise Modelling for Denoising and 3D Face Recognition Algorithms Performance Evaluation

Mehryar Emambakhsh, Jiangning Gao, and Adrian Evans,

Abstract

This paper proposes an algorithm is proposed to quantitatively evaluate the performance of 3D holistic face recognition algorithms when various denoising methods are used. First, a method is proposed to model the noise on the 3D face datasets. The model not only identifies those regions on the face which are sensitive to the noise but can also be used to simulate noise for any given 3D face. Then, by incorporating the noise model in a novel 3D face recognition pipeline, seven different classification and matching methods and six denoising techniques are used to quantify the face recognition algorithms performance for different powers of the noise. The outcome: (1) shows the most reliable parameters for the denoising methods to be used in a 3D face recognition pipeline; (2) shows which parts of the face are more vulnerable to noise and require further post-processing after data acquisition; (3) compares the performance of three different categories of recognition algorithms: training-free matching-based, subspace projection-based and training-based (without projection) classifiers. The results show the high performance of the bootstrap aggregating tree classifiers and median filtering for very high intensity noise. Also, when different noisy/denoised samples are used as probes or in the gallery, the matching algorithms significantly outperform the training-based (including the subspace projection) methods.

Index Terms

Classification, denoising, face recognition, noise modelling

M. Emambakhsh, J. Gao and A.N.Evans are with the Department of Electronic and Electrical Engineering, University of Bath, Bath, UK, BA2 7AY.

E-mail: {M.Emambakhsh, J.Gao, A.N.Evans}@bath.ac.uk

I. INTRODUCTION

Noise is an unavoidable component of all electronic devices and is present in the three-dimensional imaging systems which have been widely used for the 3D face recognition approaches in the last decade. Noise can have a significant influence on the performance of 3D face recognition algorithms. It can adversely affect the face detection and segmentation, landmarking, feature extraction and ultimately, classification. As noise is a completely random process, it changes the facial surface in a different way that it changes the corresponding gallery samples. This can result in a significant deterioration of the within-class similarity and may move a sample's feature vector into the feature space of other subjects.

To address these issues, denoising is usually applied as one of the initial steps of a 3D face recognition algorithm. However, to date, the influence of denoising on the face recognition performance has not been objectively studied. Questions such as which parts of the face are more sensitive to the noise, which denoising algorithm is the best choice for robust face recognition, how to find the optimal parameters for the denoising method and which classification algorithm performs better over a noisy dataset have not been properly analysed.

In this paper, the effects of varying the noise power on the 3D holistic face recognition algorithms are evaluated. First, instead of manually applying Gaussian noise to the face surface, as used in [1], [2], an algorithm is proposed to learn the noise distribution from the 3D faces and simulate it on any given face. The method consists of finding the Eigenshape of the difference maps computed over the aligned faces of each subject. Then, a probability map is defined and used to model the noise. The method provides the capability to gauge the effect of denoising on the performance of face recognition algorithms.

The main contributions of this work are as follows: (1) it reports a detailed quantitative evaluations of the performance of denoising methods, applied over the widely used holistic face recognition; (2) a very fast and accurate algorithm to learn and simulate the noise from 3D faces is proposed; (3) the new technique is used to identify those parts of the face which are more vulnerable to 3D reconstruction noise and (4) using the proposed approach, the most robust classification and denoising methods from 3D face recognition applications are identified.

This paper presents a continuation of our initial work which was dedicated to finding the best parameters for denoising algorithms, when used in a 3D face recognition approach [3] and,

to the best of our knowledge, is one of the most comprehensive quantitative analysis of the performance of denoising algorithms in face recognition approaches.

The paper is organised as follows. First, a brief explanation on the previous denoising approaches is provided in section II. Then, the noise learning and simulation procedures are explained in section III. A selection of the most widely used denoising and face recognition methods are described in sections IV and V, respectively. Section VI contains experimental results and conclusions are given in section VIII.

II. Recent literature review

The widespread adoption of 3D laser scanners as the imaging modality for many 3D face datasets has led to a relatively standard approach to denoising. Generally, four denoising steps are included: spike noise removal, surface smoothing, hole filling and missing data replacement. The holes and missing data differ in the sense that the depth values for the holes are available, and are usually significantly lower than for the neighbouring pixels, while the locations of missing data do not have any depth values. Spikes are a common type of noise produced by 3D laser scanners. As they usually cause impulsive variations in the face surface, median filters, the most popular approach for impulsive noise removal, are typically employed to remove them from the face's depth map [4], [5], [6], [7].

Spikes can also be interpreted as outliers in the data and, based on this assumption, statistical regional information can be used to determine the noise locations [8]. In [9], the distances between the central point and its 8 nearest neighbours are calculated. To detect the outliers, the standard deviation of the distances is then thresholded. A similar approach is used in [10], except that a neighbourhood is defined using an 11×11 mask and the angle between the optical axis and surface normal of the observed points is also used to identify the outliers. The major issue with these algorithms is their sensitivity to the mask size and thresholds used. These parameters are usually set using trial and error on a given dataset but, if the imaging modality changes, retuning is required if over- or under-smoothing is to be avoided.

Surface smoothing is required as the raw depth image usually contains artefacts caused by high-frequency components. These unwanted variations are not as salient as spike noise and are distributed over the surface. The most common approach to alleviate this type of noise is to low-pass filter the facial surface. For instance, Gaussian [1], [2], [4], [5], [8], [11], [12], [6],

[7], [13] and mean filters [14] are used to smooth and also remove the spikes from the depth surface. The median filter also has a smoothing effect and in [15] a 2.31×2.31 mm^2 median filter is applied to reduce the high-frequency noise effects. However, unlike Gaussian filters, median filtering can affect the position of image edges, which is not desirable. A different approach is to use an adaptive filtering to denoise the face surface, for example the Weiner filter [16].

The main purpose of hole filling algorithms is to detect the holes and valleys on the face surface, which are produced by inaccurate or low quality imaging, and replace them with legitimate values. Morphological filling is one of the most popular algorithms for this task [17], [18]. Applying the filling algorithm directly to the depth map can unintentionally fill some natural holes on the face, such as the eye corners and in order to avoid this issue, the difference between the original and filled surfaces is calculated and those pixels with a difference above a threshold are considered to be holes and are replaced using cubic interpolation [15].

Missing data is a common issue associated with the 3D imaging systems. It can occur due to self-occlusion (which appears after pose correction), large depth variations (for instance, due to open mouths or the nostrils) or imaging device inaccuracy. These points are usually predefined as invalid points, making it easy for the users of the datasets to detect them. Also, in some denoising methods the noisy points are intentionally labelled as missing data and subsequently replaced [15], [9]. For all these cases, the most extensively used approach is to replace the missing data using interpolation. The main interpolation methods used are: cubic [9], [12], bicubic [8], [19], linear [10], [16], [14], [20] and K-nearest neighbour interpolation (KNN) [21].

III. Noise modelling using a probability map

The noisy samples in the face datasets can be used to find a model for the noise which can then be used to simulate noise over other, less noisy samples. First, it is assumed that the i^{th} subject ($i = \{1, \ldots, N\}$) in the dataset has J_i non-occluded samples with neutral facial expression. These samples are resampled using the Delaunay triangulation to interpolate the missing data and remove the noise in the coordinate maps. This operation results in a set $\{\mathbf{F}_{i,1}, \mathbf{F}_{i,2}, \ldots, \mathbf{F}_{i,J_i}\}$ where each $\mathbf{F}_{i,j} = [\mathbf{X}_{i,j}, \mathbf{Y}_{i,j}, \mathbf{Z}_{i,j}]$ is a $M \times 3$ matrix whose columns correspond with the x, y and z axes points. Then, using a 2.5×2.5 mm^2 mask, a median filter is applied twice to remove the spike noise and smooth the surface. Following [9], the faces are then aligned using singular value decomposition over the 3×3 covariance matrix $\Sigma_{i,j}$ for the j^{th} sample of the i^{th} subject

by,

$$\mathbf{V}_{i,j}^{-1}\Sigma_{i,j}\mathbf{V}_{i,j} = \mathbf{E}_{i,j}, \tag{1}$$

where $\mathbf{V}_{i,j}$ and $\mathbf{E}_{i,j}$ are the 3×3 eigenvector and eigenvalue matrices, respectively. $\mathbf{V}_{i,j}$ is used as a rotation matrix and multiplied by the noisy input point clouds after their averages, $\mathbf{m}_{i,j}$ (a 1×3 vector), are removed. The result is the aligned set of point clouds, $\mathbf{F'}_{i,j}$, given by

$$\mathbf{F'}_{i,j} = (\mathbf{F}_{i,j} - \mathbf{I_o}\mathbf{m}_{i,j})\mathbf{V}_{i,j}, \tag{2}$$

where $\mathbf{I_o}$ is a $M \times 1$ column vector, whose elements are one. After this operation, all the J_i noisy samples of the i^{th} subject are aligned. As $\mathbf{F'}_{i,j}$ will be the depth map independent from the resolution information, its first and second columns, which contain the x and y axes data, are discarded. This makes $\mathbf{F'}_{i,j}$ an $M \times 1$ vector.

The aligned faces are all resized to a fixed height and width and the pair-wise pixel-by-pixel differences between the aligned faces of each subject are computed and accumulated resulting in difference maps for each subject ($\circ$ is the Hadamard's component-wise product),

$$\mathbf{D}_i = \sqrt{\sum_{m,n,m\neq n} (\mathbf{F'}_{i,m} - \mathbf{F'}_{i,n}) \circ (\mathbf{F'}_{i,m} - \mathbf{F'}_{i,n})}. \tag{3}$$

where $\mathbf{D}_i$ is the difference map for the i^{th} subject. This procedure is repeated for all the subjects in the dataset, giving the difference matrix $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N]$. In order to find a map containing the maximal shape variation between the difference maps, the PCA algorithm is applied to $\mathbf{D}$ and the first eigenvector, corresponding to the highest eigenvalue is obtained. By performing this PCA procedure the N-dimensional space is projected into one dimension, in which the maximum variance of the data distribution is preserved, by

$$\begin{cases} \mathbf{V}_{\mathrm{D}}^{-1}\Sigma_{\mathrm{D}}\mathbf{V}_{\mathrm{D}} = \mathbf{E}_{\mathrm{D}} \\ \bar{\mathbf{D}} = (\mathbf{D} - \mathbf{I_o}\mathbf{m}_{\mathrm{D}})\mathbf{U} \end{cases} \tag{4}$$

where Σ_{D} is an $N \times N$ matrix covariance matrix and $\mathbf{m}_{\mathrm{D}}$ is a $1 \times N$ vector containing the average of the rows of $\mathbf{D}$ and $\mathbf{U}$ is a column vector including the eigenvector corresponding to the highest eigenvalue, which is obtained from $\mathbf{V}_{\mathrm{D}}$. $\bar{\mathbf{D}}$ is the "eigen-"difference shape, which

includes the highest variances of $\{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N\}$. $\bar{\mathbf{D}}(x, y)$ shows the significance of the noise strength at point (x, y). If the area around (x, y) is flat, then the noise power will be very low over the area. Alternatively, peaks or valleys in $\bar{\mathbf{D}}$ correspond to a higher probability of having spikes or holes in that particular region of the face. Since all the images used to create $\bar{\mathbf{D}}$ have a neutral expression, are non-occluded and aligned, the only difference between them is the noise. Consequently, the values of $\bar{\mathbf{D}}$ at different points provide a good measure to identify which parts of the face are noisier after the 3D data capture.

A major, additional value of $\bar{\mathbf{D}}$ is that it can be used to create a probability map which can be used to artificially simulate noise over any set of 3D face data. This will enable the noise power to be stochastically varied thus underpinning the quantitative evaluation of both the denoising performance and the robustness of face recognition algorithms. The noise occurrence probability map, $\mathbf{P}$, for each pixel (x, y), is simply found by normalising $\bar{\mathbf{D}}$ to the range 0 to 1, using the min-max normalisation,

$$\mathbf{P} = \frac{\bar{\mathbf{D}} - \min_{x,y}(\bar{\mathbf{D}})}{\max_{x,y}(\bar{\mathbf{D}}) - \min_{x,y}(\bar{\mathbf{D}})} \tag{5}$$

Any face data in the dataset can now be degraded by adding a randomised simulated noise map $\delta\mathbf{F}$ to its depth image by,

$$\begin{cases} \mathbf{F}^n_{i,j} = \mathbf{F}_{i,j} + \delta\mathbf{F}_{i,j} \\ \delta\mathbf{F}_{i,j}(x, y) = \begin{cases} a\bar{\mathbf{D}}(x, y), & \text{if } \mathbf{P}(x, y) \geq \mathbf{r}_{i,j}(x, y) \\ 0, & \text{if } \mathbf{P}(x, y) < \mathbf{r}_{i,j}(x, y) \end{cases} \end{cases} \tag{6}$$

where $\mathbf{r}_{i,j}$ is a matrix the same size as $\mathbf{P}$, whose elements are randomly assigned, for the j^{th} sample of the i^{th} subject, using a uniform distribution in the range $[0, 1]$. $\mathbf{F}^n_{i,j}$ is the noisy face image and a is a scalar used to vary the noise power. When $\mathbf{P}(x, y)$ is high (e.g. at fragile, noisy parts of the face), its value will be more likely to be higher than the uniformly selected random number $\mathbf{r}_{i,j}(x, y)$ and the noise at that point is amplified. On the other hand, lower values of $\mathbf{P}(x, y)$ (which correspond to less noisy facial parts after 3D reconstruction), reduce the probability that $\mathbf{P}(x, y) \geq \mathbf{r}_{i,j}(x, y)$ and hence of additional noise being added to the data at that point. The whole procedure for finding the eigen-difference shape $\bar{\mathbf{D}}$ is depicted in Fig. 1.

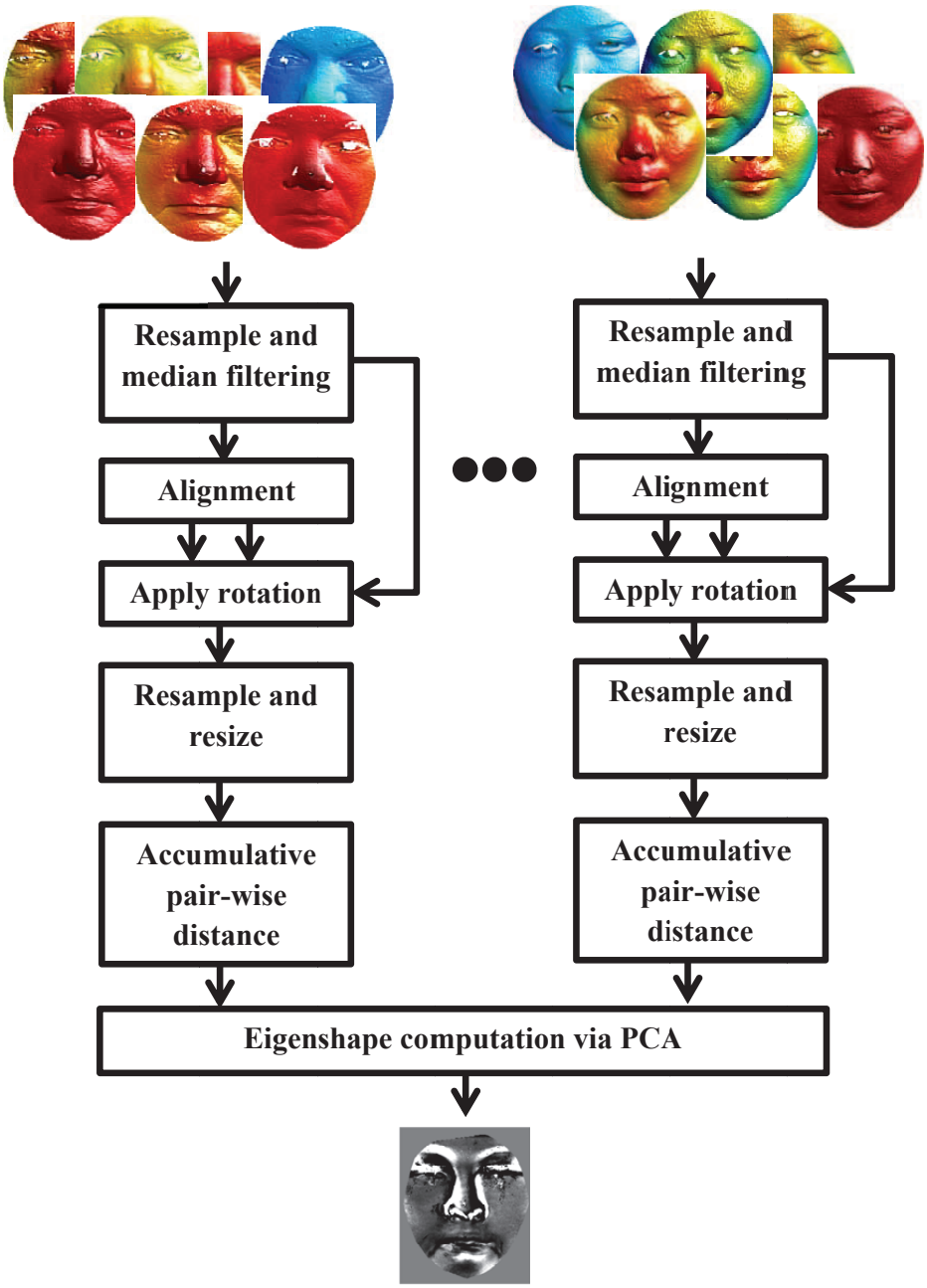


Fig. 1: The procedure of finding the eigen-difference shape $\bar{\mathbf{D}}$: After resampling and alignment, the mean vectors and pose rotation matrices are computed per sample, which are then applied over the input noisy point clouds to compute accumulative difference map $\mathbf{D}_i$. $\bar{\mathbf{D}}$ is eventually calculated over $\mathbf{D}_i$.

IV. DENOISING ALGORITHMS

Six denoising algorithms are evaluated using the dataset: Gaussian, mean and median filtering, multi-scale wavelet denoising, adaptive Wiener filtering and non-linear diffusion. The denoising algorithms are selected because they have been widely used in previous 3D face recognition algorithms and also to cover the diversity in the methodologies for reducing the effects of noise. Median filtering is capable of successfully removing non-linear noise, such as spike and impulsive noise, while linear filters, such as mean, Gaussian and wavelets can better filter high frequency noise. While both these linear filters can smear the edges in the images, non-linear diffusion has a significantly better edge preserving capability. All these methods are deterministic and filter without utilising statistical information. However, as an adaptive stochastic filter, Weiner denoising estimates the signal and noise distributions over a given mask size. The noise is then removed by sampling from the estimated distributions. The denoising methods and their parameters are briefly explained below. For all filters, the masks sizes are given in pixels as the faces are resampled using a uniform grid with a 0.5 mm/pixel horizontal and vertical resolution.

1) Gaussian, mean and median filtering: The large variations in the depth map are concentrated in its high-frequency components. Low-pass filtering is the most common way to remove these components, for example by convolving Gaussian filters ($\mathbf{G}_{\sigma}^{M}$) with the image, where σ is the standard deviation and M is the square root of the mask size. Similarly, the mean filter can also be used to reduce the high-frequency noise and smooth the face surface. For both filters, the size of the mask is varied and the recognition results recorded. The median filter has also been extensively used for preprocessing 3D faces and a similar approach is employed to evaluate its denoising performance.

2) wavelet filtering: Wavelet decomposition is one of the standard methods used for the 1D and 2D signals denoising. L levels of the multi-resolution wavelet decomposition are utilised in our work. Using the results obtained in [3], various wavelets are applied, including Daubechies, Coiflets, Symlets, discrete Meyer, Biorthogonal and Reverse Biorthogonal, with the results showing the superiority of the discrete Meyer wavelet. The default thresholds (soft or hard thresholds) are calculated using the seminal algorithm proposed by Donoho and Johnstone ([22]), and the decomposition is performed in L levels.

3) Weiner filtering: Weiner filtering is a type of adaptive denoising algorithm, which uses the statistical information of the input image [23]. An $M_w \times M_w$ neighbourhood is used to estimate

the noise's statistical parameters, such as variance and mean. By varying M_w, the aggressiveness of the denoising is changed and a wider image area is analysed.

4) Non-linear diffusion: Non-linear diffusion is a method introduced by Perona and Malik for image simplification, denoising, segmentation and feature extraction [24]. Its concept is based on the heat transmission between adjacent materials. The partial differential equation of diffusion is iteratively solved over the image domain. If the parameters are appropriately tuned, the result is a denoised version of the input image. The most interesting feature of the non-linear diffusion is its edge preservation, while smoothing the adjacent regions. This means that, for example, high-frequency noise can be removed from the forehead while the edges close to the eyes are maintained. Following [25], the diffusion equation used here is,

$$\frac{\partial \mathbf{Z}}{\partial t} = \nabla.(g_m(|\nabla \mathbf{Z}|^2)\nabla \mathbf{Z}), \tag{7}$$

in which $\mathbf{Z}$ is the depth image of the face, $\nabla.()$ and ∇ are the divergence and gradient operators, respectively, and $g_m(.)$ is a decreasing function [25] given by,

$$g_m(s) = \begin{cases} 1 - \exp(-(\frac{\lambda_m}{s})^m), & \text{if } s > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where λ_m is the image contrast control parameter in the m^{th} iteration of the diffusion equation. As λ_m increases, the resultant diffused image is more significantly blurred.

V. Face recognition methods evaluation pipeline

The evaluation employs holistic 3D face recognition algorithms, which have been widely used for both 2D and 3D face recognition. In the first step, all faces are cropped and aligned. As the noise in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps is much less marked than that in the $\mathbf{Z}$ depth map [9], [15], [26], here the denoising is only applied to the depth image. To perform this task, after acquiring the 3D raw data, resampling is applied to replace the noise and missing data in the $\mathbf{X}$ and $\mathbf{Y}$ coordinate maps. The nose tip is relocalised on the resampled data and, after temporarily replacing the depth value of the invalid points with the median of the valid points' depth, the denoising algorithm is applied. Then, a sphere, centred on the nose tip, with radius 80 mm is then intersected and the facial region is cropped.

The depth maps are resized to the same size as the eigen-difference shape $\bar{\mathbf{D}}$ and the noise power is tuned using a in (6). Simulated noise is added to the depth maps and then the denoising algorithms described in the previous section are applied. The parameters of the denoising algorithms are set to those that gave the best 3D face recognition performance in [3], in which first, the faces are resampled on a uniform grid, then different denoising algorithms are applied. The faces are then cropped, aligned and normalised. The resulting faces are concatenated to form the feature space. By changing the parameters of the denoising algorithms, various feature spaces with different class Separabilities are obtained. The outcome is sets of extensive experimentations showing that performing a more aggressive denoising algorithm (having larger masks, wider low-pass frequency responses and smoothing effects) does not necessarily decrease the face recognition performance [3]. In addition to this, it is found that optimal parameters can be found for some of the denoising algorithms, which maximise the recognition rates. This procedure is illustrated in Fig. 2-a.

After applying the optimal denoising methods, the depth values are normalised using the min/max normalisation of (5) and the resulting map is again resized. For all the resizing steps, the face is pre-filtered to avoid the high frequency aliasing and cubic interpolation is used for resampling. Finally, the resulting samples are divided into gallery and probe samples and the resulting feature vectors used by the classification or matching algorithms. The stages of the recognition pipeline are illustrated in Fig. 2-b, for two example faces.

The face recognition pipeline of Fig. 2 enables the robustness evaluation of different face recognition algorithms against varying noise powers. This approach also quantitatively identifies the best performing denoising algorithms, in terms of recognition performance.

To evaluate the denoising algorithms seven widely used classification algorithms in the field of holistic face recognition are used: multi-class Support Vector Machines (m-SVM), PCA, Kernel Fisher's Analysis (KFA), Probabilistic Neural Network (PNN), KNN-classification, bootstrap aggregation decision trees (TreeBagger) and Linear Discriminant Analysis (LDA). For the m-SVM classifier a linear kernel is used and the one-vs.-all scenario is utilised to transform it to a multiple-class classifier. For the subspace projection approaches [PCA (Eigenfaces [27]), LDA (Fisherfaces [28]) and KFA (a kernel-based extension of Fisherfaces)] the feature space is projected onto a 100-dimensional space. The polynomial kernel is employed for the KFA algorithm and the city-block (L1 norm), Euclidean (L2 norm) and cosine distances are used for

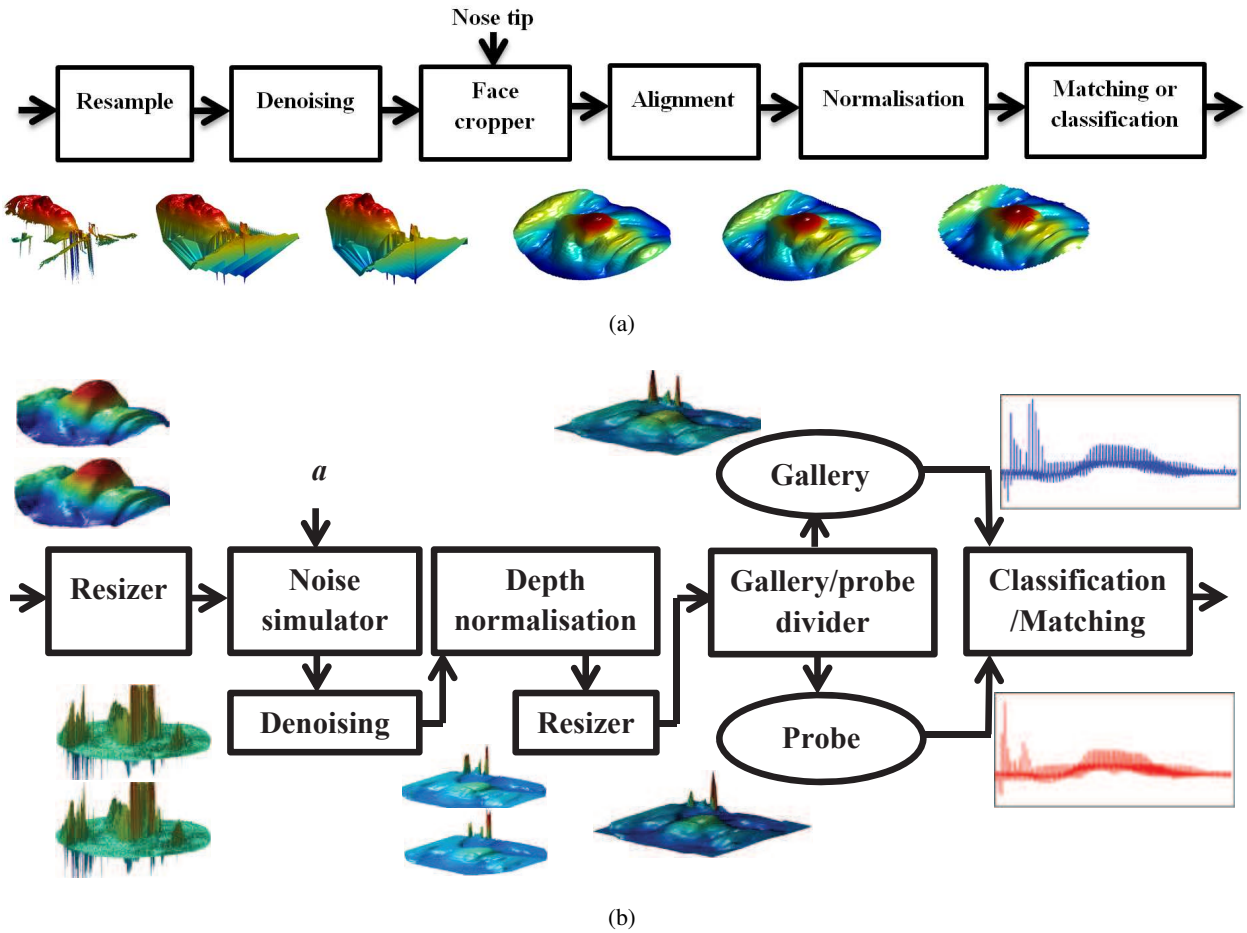


Fig. 2: (a) The face recognition pipeline used to find the optimal parameters for the denoising algorithms from [3]. (b) The 3D face recognition pipeline, including the noise simulation procedure ($a = 0.5$ for the simulated noisy images). After resizing, noise is simulated over the input depth maps using a in (6). Then the denoising algorithm is applied and the feature vectors are created after normalisation and resizing.

the KNN classifier ($K = 2$). Finally, 119 trees are aggregated to create the TreeBagger classifier [29]. These approaches cover a wide range of classification methods and have been extensively used in for holistic 3D face recognition. PCA performs unsupervised, low dimensional projection while LDA, KFA, PNN and SVM classify samples by linear or non-linear mapping to either lower (LDA, KFA and PNN) or higher (SVM) dimensions using a supervised approach. A multi-classifier classification is performed by TreeBagger, whose bootstrap aggregation feature makes

it superior to the AdaBoost algorithm, which is sensitive to the absence of sufficient training samples per subject.

All of the above approaches are examples of discriminative classification algorithms, which estimate the posterior class probability, as opposed to the generative models which estimate the joint class and observations probabilities. As shown by Ng *et al.*, discriminative models have better performances than the generative models when there are not enough samples per classes, such as realistic biometric applications [30]. Generative models, which estimate the joint probability distributions of the observations and classes, are significantly more robust when a very large dataset is available, making them very useful for text processing and big data algorithms [31], [30]. This is the reason why they have not been evaluated in our work.

VI. Experimental results

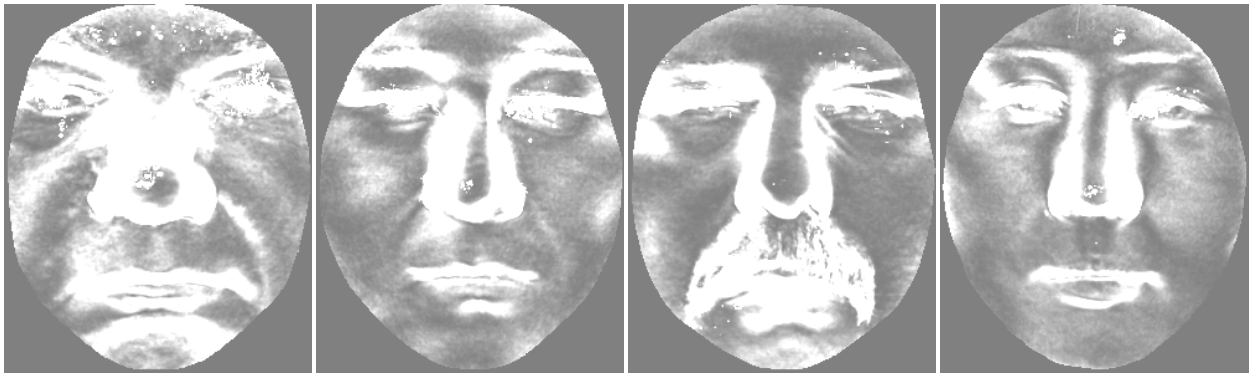
A. Dataset configuration

The FRGC dataset is used in our work to evaluate the proposed algorithm. It is one of the largest publicly available 3D face datasets, in terms of the number of subjects. Overall, FRGC contains 557 subjects, whose samples are obtained in three different time periods: Spring 2003, Fall 2003 and Spring 2004, captured using the Minolta Vivid 900/910 laser sensor, which is also utilised by other widely-used face datasets [32], [33]. Unlike more recent datasets such as the Bosphorus [33] and the UMB-DB [32], samples of the FRGC dataset are not postprocessed and hence its data contains all the different types of noises in 3D faces: spike, high frequency noise, holes and missing data.

The data in the Spring 2003 folder is significantly noisier than that for the other seasons, probably due to older capture devices being used. The noise intensity in the samples in this folder makes it more appropriate to evaluate the denoising algorithms robustness and therefore the folder is utilised in all the following experiments in this paper. In particular, this folder is used to find the probability map of the noise, hence identifying those parts of the face that are more vulnerable to noise after the 3D reconstruction. The denoising algorithms performance is also quantitatively evaluated using this folder. Those classes with at least four samples in the folder are used, giving a dataset of 119 subjects (classes) and 661 samples. Two samples per class are selected as the training (gallery) and the remaining samples as probe images.

B. Noise model computation

Those subjects from the FRGC 2003 folder with at least seven samples are used to calculate difference maps $\mathbf{D}_i$. This results in a subset consisting of 28 subjects with 209 samples in total. Example shape difference maps $\mathbf{D}_i$ for four subjects are shown in Fig. 3-a to -d, in which the regions that are more different from the neighbouring pixels have higher grey scale values. These differences are caused by the noise, since the captures used have neutral expressions and no occlusions. It is interesting to see that regions located on the eyebrows, eyes, surrounding nasal region and mouth have the highest vulnerability to the noise, as the within-class similarity is at its lowest on these parts. The high frequency of the depth variations over these regions has caused higher errors in the 3D reconstruction while the flatter parts of the face, such as the forehead, cheeks and chin, are less noisy.

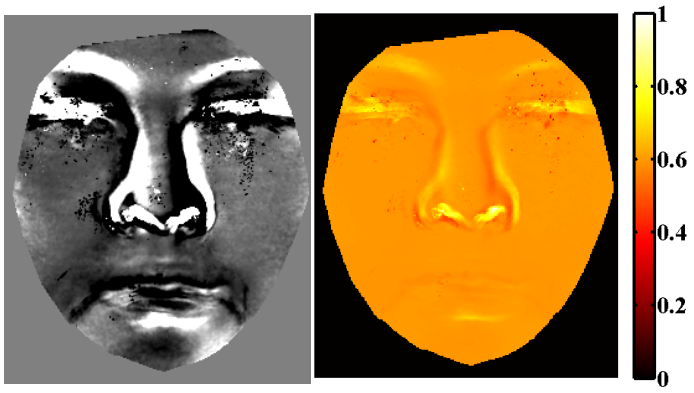


(a) (b) (c) (d)

Fig. 3: Shape difference maps for four different subjects: $(\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4)$.

The eigen-difference shape ($\bar{\mathbf{D}}$) computed over all the difference maps is plotted in Fig. 4-a. $\bar{\mathbf{D}}$ contains the maximal shape variations among the difference maps $\mathbf{D}_i$ and shows that the nostrils, nose tip and sides are very sensitive to the reconstruction noise of the 3D scanner. These are the regions where the influence of the denoising methods will be more obvious. In particular for the nose tip, which is used for face segmentation, inaccurate denoising can lead to incorrect facial region cropping.

When computing the eigen-difference shape $\bar{\mathbf{D}}$, if the number of samples per subject is too small, the resulting difference maps ($\mathbf{D}_i$) will not be accurate enough. Alternatively if it is too

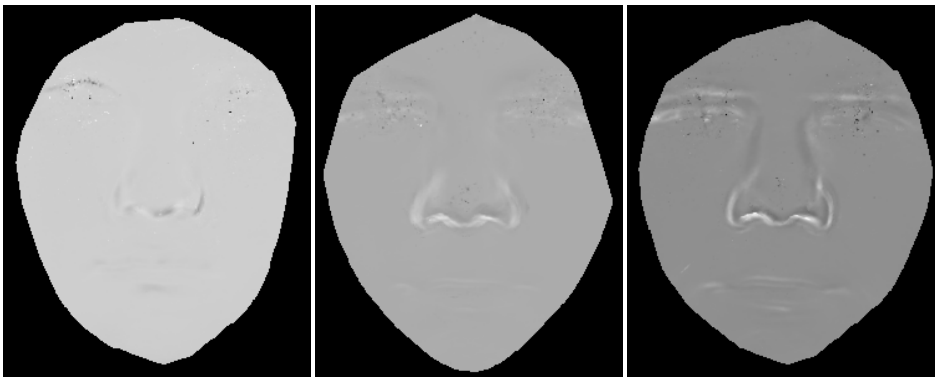


(a) (b)

Fig. 4: (a) The eigen-difference shape result ($\bar{\mathbf{D}}$), and (b) the probability map ($\mathbf{P}$).

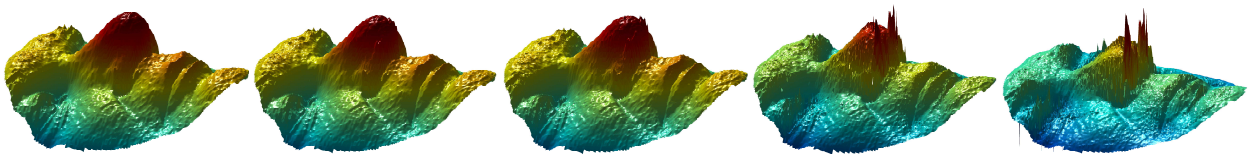
high, as there are not many samples per subject for most of the subjects of the dataset, there will be an insufficient number of subjects to be able to accurately find $\bar{\mathbf{D}}$. Figure 5 shows $\bar{\mathbf{D}}$ computed using different numbers of samples per subjects. When the number is small, $\mathbf{D}$ is unable to signify the within class dissimilarities caused by the noise and it is also more sensitive to small alignment errors. This increases the variance of the feature space scatter and increases the range of PCA lower dimensional projection. The resulting $\bar{\mathbf{D}}$ will be a relatively more uniform image, incapable of representing spike noise, see Fig. 5-a. Increasing the training samples can produce richer difference maps $\mathbf{D}_i$, in the sense that more information about the regional noise is provided. This can be seen in Fig. 5-c around the nostrils, eyes, eyebrows and nasal tip. These considerations underpin the selection of seven samples per subject in the experiments to compute $\bar{\mathbf{D}}$.

Normalising $\bar{\mathbf{D}}$ using (5) results in the probability map ($\mathbf{P}$) shown in Fig. 4-b. Those regions of $\mathbf{P}$ which have higher values are more likely to be affected by noise. Using $\mathbf{P}$ in (6) to simulate noise on the aligned depth map produces the simulated. Figure 6-a shows an example of the aligned depth map and Fig. 6-b to -e show the noisy images as the noise power is increased by varying a in (6). As the noise power is intensified, the spike, holes and high frequency noise will become more salient in the depth maps.



(a) (b) (c)

Fig. 5: The eigen difference shape $\bar{\mathbf{D}}$ when the number of samples per subjects is: (a) 3 (166 subjects), (b) 4 (119 subjects) and (c) 5 (86 subjects). As the number is increased, the noisy areas become more salient, resulting in a more accurate noise modelling.



(a) (b) (c) (d) (e)

Fig. 6: (a) An example of the input depth map $\mathbf{F}_{i,j}$ and the resulting $\mathbf{F}^n_{i,j}$ using (b) $a = 0.0005$; (c) $a = 0.001$; (d) $a = 0.005$ and (e) $a = 0.01$ in (6).

C. Performance of denoising methods

The use of noise modelling to simulate the noise over different depth maps enables a quantitative evaluation of the success of different denoising and classification/matching algorithms to be performed. To this end, the seven widely-used denoising methods from section V are used as part of a face recognition algorithm to recognise faces. The recognition algorithms from section V are the m-SVM, PCA, KFA, PNN, TreeBagger, LDA and KNN classifier, with three different distance criteria: cosine (Cos), Euclidean (Euc) and city-block (Ctb). Matlab's Statistics and Neural Networks Toolboxes are used to implement the m-SVM, TreeBagger and PNN classifiers, while the PhD (Pretty helpful Development functions for face recognition)

toolbox [34] is utilised to perform the subspace projection (PCA, LDA and KFA) and KNN classification. For the implementation of the nonlinear diffusion filter, the "Nonlinear Diffusion Toolbox" provided by F. D'Almeida is employed [35].

Denoising methods	**Parameters**
Gaussian filtering	$M = 37$ mask and $\sigma = 6$
Median filtering	33×33 mask
Mean filtering	19×19 mask
Non-linear diffusion	Exp4
Weiner filtering	$M_w = 19$
Wavelet filtering	Five levels ($L = 5$) of the discrete Meyer wavelet decomposition

TABLE I: The configuration for the denoising algorithms (Exp4 is explained in [3]).

The results of applying the discrete Meyer wavelet, mean, Gaussian, median and Weiner filtering, and non-linear diffusion over four different noise powers are plotted in Table II. Four different noise powers are utilised: $a = 0, 0.5, 1, 1.5$. $a = 0$ corresponds to the case in which no noise is simulated on the input depth image. The parameters of the denoising algorithms are selected from the best results reported in [3] and are described in Table. I. For each classification algorithm, the rank-one recognition rate is calculated. For almost all cases, increasing the noise powers reduces the recognition performance, as expected.

When the noise power is low ($a \approx 0$), the KFA face recognition approach has the best performance for all denoising approaches (shown in red in Table II). However, as the noise power increases, median filtering followed by the TreeBagger, m-SVM, KFA, PNN and KNN-Ctb classifiers produce the highest recognition ranks. For instance, for the m-SVM classifier (shown in cyan in Table II), median filtering produces significantly higher rank-one recognition rates compared to the other denoising techniques. A similar trend exists for the other classifiers, shown in magenta in Table II for the median filtering denoising.

In contrast, for the subspace projection-based classification methods, which are PCA, KFA, PNN and LDA, Weiner filtering generates the highest recognition ranks, when the noise power is high for example $a = 1.5$; this is displayed in green in the table. When the noise power is high, the KNN-Ctb and TreeBagger result in the highest rank-one rates. This is denoted in blue

Algorithm	Noise power	m-SVM	PCA	KFA	PNN	KNN-Ctb	Tree-Bagger	LDA	KNN-Euc	KNN-Cos
Discrete Meyer	$a = 0$	93.83	94.68	97.66	93.62	94.47	93.83	95.53	90.43	84.47
	$a = 0.5$	63.40	62.98	61.70	62.77	85.11	80.00	62.55	62.77	62.98
	$a = 1$	61.91	61.28	61.49	61.70	70.85	71.91	60.85	61.70	61.91
	$a = 1.5$	62.13	61.49	61.91	61.49	68.30	68.94	61.06	61.70	61.49
Mean filtering	$a = 0$	96.38	95.32	99.36	94.04	93.83	93.62	97.23	91.70	84.68
	$a = 0.5$	74.47	75.32	80.00	71.91	89.57	80.09	73.83	70.64	65.32
	$a = 1$	66.38	67.23	71.28	63.62	82.77	77.87	65.53	66.60	57.45
	$a = 1.5$	63.62	64.89	68.51	63.83	73.62	78.09	64.68	62.77	52.55
Gaussian filtering	$a = 0$	95.32	95.11	98.94	92.55	92.98	94.47	97.66	90.00	81.70
	$a = 0.5$	73.62	78.51	83.62	72.34	87.23	84.89	83.40	72.55	69.57
	$a = 1$	67.87	67.45	75.32	65.96	80.64	76.81	70.21	65.74	62.13
	$a = 1.5$	64.04	65.74	72.98	63.19	73.62	73.40	69.15	62.13	59.57
Non-linear diffusion	$a = 0$	94.26	94.26	97.02	92.34	93.83	94.47	92.34	89.79	81.91
	$a = 0.5$	63.62	63.83	63.62	61.91	86.60	82.98	61.28	63.19	63.40
	$a = 1$	61.49	61.28	61.70	61.49	68.94	78.72	60.85	61.28	61.06
	$a = 1.5$	61.06	60.85	61.28	61.49	65.32	75.32	61.06	60.85	61.28
Weiner filtering	$a = 0$	95.32	94.89	98.51	92.34	93.19	91.28	96.81	90.00	81.70
	$a = 0.5$	70.43	75.96	83.40	72.13	86.60	83.62	78.51	70.85	63.40
	$a = 1$	64.26	70.43	77.87	69.15	77.66	74.04	71.28	65.74	57.45
	$a = 1.5$	63.62	68.72	77.45	68.09	74.26	72.77	68.94	65.96	56.60
Median filtering	$a = 0$	95.74	94.47	98.30	92.98	93.40	93.83	96.38	90.85	83.40
	$a = 0.5$	89.15	75.11	85.32	74.68	92.77	90.64	79.79	60.00	46.81
	$a = 1$	80.64	67.23	75.96	69.15	84.89	90.00	72.77	52.77	40.85
	$a = 1.5$	75.11	62.77	72.55	65.96	81.28	89.36	69.15	48.30	35.96

TABLE II: Rank-one recognition rates (in %) for different noise powers and denoising algorithms. The high KFA performance for low noise levels is shown in red, while the robustness of the KNN-Ctb and TreeBagger against noise in high noise powers is marked in blue. The high potential of the median filtering to denoise faces is signified in magenta and cyan, while the higher performance of the Weiner filtering when used prior to subspace projection methods is shown in green.

for the highly intensive noise power ($a = 1.5$), for the KNN-Ctb and TreeBagger in Table II.

To summarise the results, although the performance varies for each classification algorithm, the median and Weiner filters generally result in the highest recognition ranks. Despite their smoothing effects, non-linear diffusion, discrete Meyer wavelet and mean filtering produce the lowest average recognition ranks, which verifies their inability to completely remove the impulsive spike noise. The lowest classification results are achieved when the cosine and Euclidean distances are applied with the KNN classifier.

Table II also shows that the performance of the subspace projection methods (PCA, KFA and LDA) significantly deteriorates as the noise level increases. For example, the recognition ranks for the KFA classification method drop by $\approx 27\%$ as noise level is increased from $a = 0$ to 1.5. This is because of the sensitivity of these methods to the outliers in the data that occurs more

frequently at high noise powers.

To better compare the performances of the classification methods, a is increased from 0 to 2.75 and the rank-one recognition rates are computed when median filtering is applied as the denoiser. The results, plotted in Fig. 7, verify the results in Table II, showing the KFA and LDA classifiers produce the highest recognition ranks when the noise power is low. However, their performance is slightly decreased for higher noise powers.

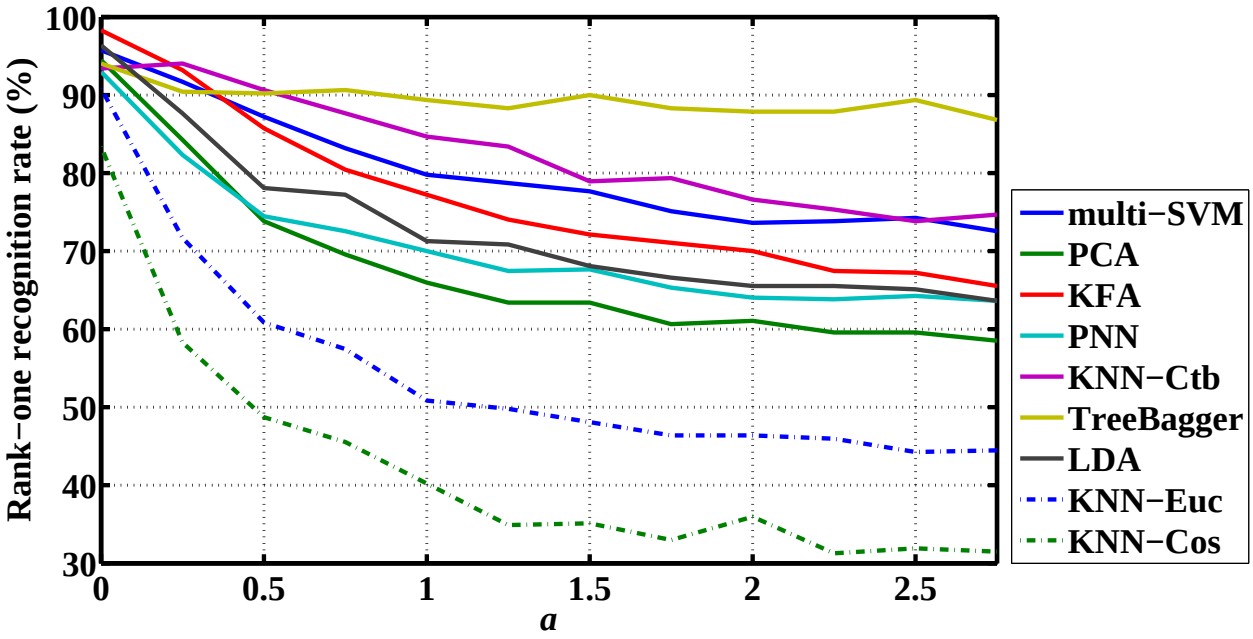


Fig. 7: Rank-one recognition rates achieved by the classifiers from Table II after median filtering denoising for different noise powers $a = 0$ to 2.75.

The TreeBagger classifier is the most robust recognition algorithm, producing the best performance for $a > 0.5$. Its output recognition rate remains very close to 90%, even at high level of noise. Among the different matching criteria the city-block distance significantly outperforms the Euclidean and cosine distances. This might be due to the higher performance of the L1-norm, when applied on a sparse feature space [15], [36], [37]. Also, the least absolute deviation matching performed by L1-norm is more robust against outliers and noise in the data than the least Euclidean deviation computation (L2-norm) [38]. The L1-norm has a built-in feature selection capability [39] and can significantly better represent sparsity than other Lp-norms $p > 1$

[21], [36], [40], [41].

D. Noise/denoised gallery vs. noise/denoised probe

Noise is a stochastic process and randomly changes the depth map. Its distribution might also change for different image acquisition devices and classification algorithms should be robust against these variations. As the final experiment, the effects of using noisy and denoised samples for the gallery and probe are evaluated. The purpose of this experiment is to quantify the performance of the face recognition algorithms when the training or test samples are either denoised or degraded by noise.

The result of applying this procedure over the Spring 2003 samples is illustrated in Table III for $a = 0.25$ and using median filtering to denoise the captures. The four classification methods which produced the highest recognition rates in the previous section are used: TreeBagger, KNN-Ctb, m-SVM and KFA. When a noisy gallery is used, the noisy probe images are recognised with rank-one recognition rates $\approx 93\%$, with the KNN-Ctb matching algorithm outperforming the training-based methods. Also, as expected, the recognition rates when both the gallery and probe samples are denoised is high, with the subspace projector KFA classifier producing a 98.30% rank-one recognition rate.

This experiment also shows how the classification methods can fail when the samples in the probe (or gallery) have different noise distribution. The classification performance significantly decreases when a noisy gallery is used for denoised probes. A similar situation exists for the case of a denoised gallery and noisy probes. Failure to detect the correct between-class boundaries or subspace projection axes are the main causes of this deterioration. Also, when the probe is noisy and gallery is denoised, the learned classes do not fit to the unseen noise in the data and under-fitting occurs. As a result, the samples are wrongly classified and recognition ranks decrease. The KNN-Ctb distance classification still has a significantly higher recognition performance than the other leaning-based approaches, producing a 81.91% rank-one recognition rate for the denoised gallery when used for the noisy probe samples.

In many practical cases, the probe faces are obtained using different imaging modalities from the gallery samples or sometimes laser scanners with different noise distributions are used. In such cases, the results in Table III, show that the direct matching algorithm KNN has a significantly higher performance than the training-based methods (TreeBagger, SVM and KFA).

Gallery	**Probe**	
	Noisy	**Denoised**
Noisy	TreeBagger: 91.91% KNN-Ctb: 94.04% m-SVM: 91.70% KFA: 93.19%	TreeBagger: 12.55% KNN-Ctb: **45.53%** m-SVM: 1.70% KFA: 10.00%
Denoised	TreeBagger: 15.32% KNN-Ctb: **81.91%** m-SVM: 2.34% KFA: 11.06%	TreeBagger: 93.83% KNN-Ctb: 93.40% m-SVM: 95.74% KFA: 98.30%

TABLE III: Noisy/denoised gallery vs. Noisy/denoised probe rank-one recognition results, when $a = 0.25$ and median filtering is used for denoising. KNN, which is a matching algorithm, outperforms other classifiers when applied over noisy probe samples to match with denoised gallery images.

VII. SUMMARISATION OF THE EXPERIMENTAL RESULTS

A. Linear denoising

Mean, Gaussian and discrete Meyer wavelets perform linear filtering over the noisy faces and are capable of removing high frequency noise from the images. As the noise power is increased the performance of the Gaussian denoising algorithm significantly outperforms the other two approaches for, all the classification algorithms. This is thought to be mostly because of the weighting of the central pixel of the Gaussian filtering mask, which can more accurately preserve edges. The mean filtering algorithm, on the other hand, smears the edges and reduces the between-classes dissimilarities.

B. Non-linear denoising

Spike noise, which is a type of impulsive noise, is randomly added to the image. Since this type of noise is spread over all of the frequency components, linear filtering algorithms can only create blobs over the noisy regions. In contrast, non-linear filtering algorithms can replace the noisy regions using a non-linear approximation of the neighbouring pixels. Median filtering and non-linear diffusion can effectively perform this task. Median filtering produces more separable classes than non-linear diffusion and it is evidenced by the median filtering $\approx 3\%$ higher recognition rates.

C. Statistical denoising

The Weiner filter is a stochastic adaptive filter, which estimates the distribution of the noise, instead of evaluating its frequency response. Therefore, given a suitable mask size, it can find the density of the signal and noise distributions and remove the noise by sampling from the estimated distribution. The result of Weiner filtering denoising outperforms the linear filters and in particular in high power of noise, significantly more robust against outliers.

D. Subspace projection-based algorithms

Amongst the different subspace projection-based algorithms evaluated in this work (PCA, LDA, KFA, PNN and m-SVM), KFA has the highest (average) recognition rate when applied to all the denoising algorithms for all noise powers. This shows that the classes are more separable when are non-linearly projected to lower dimensions than linear projections.

E. Direct matching algorithms

The KNN classification using the city-block distance outperforms the other two distance criteria. This is because of its superiority in modelling sparsity, its built-in feature selection capability and its robustness against noise and outliers in the data. The city-block distance produces, on average, $\approx 12\%$ and $\approx 18\%$ higher recognition rates than the Euclidean and cosine distances, respectively.

F. Multi-classifier algorithms

The TreeBagger algorithm shows the highest robustness against noise. This is because of its capability to sample new training data from the given observations and then combine several tree classifiers. This feature of the TreeBagger algorithm makes it superior to a similar tree-based classifier combination using the AdaBoost algorithm, which is sensitive to the absence of enough training samples per subjects.

VIII. Conclusion

This paper explores the robustness of denoising and 3D holistic face recognition algorithms for different noise powers. To be able to quantitatively evaluate the robustness of different classification and denoising methods, the novel approach of learning the noise distribution over the facial surface and then simulating it over other samples is proposed. The 3D face recognition evaluation pipeline is used to evaluate the denoising techniques of non-linear diffusion, median, mean, Gaussian, Weiner and wavelet filtering, applied before seven different classification methods, including SVM, neural networks, Tree-based, KNN matching and subspace projection methods.

Median, Gaussian and Weiner filtering generate the best results, with the median filter producing the best overall classification performance for high intensity noise. For low intensity noise, the subspace projection classifiers (KFA and LDA) are the best performing classifiers. However, when the noise intensity is increased, the performance of subspace projection methods significantly deteriorates and the experiments show the TreeBagger and KNN with the city-block distance to have the best robustness. The use of denoising/noisy samples for the gallery/probe is also evaluated and the results show that the matching algorithms, for example KNN-Ctb, significantly outperforms the training-based methods, as they do not rely on classification boundary allocation, feature space mapping or subspace projection.

A. Future work

The proposed method to estimate the noise distribution and applying over faces is completely database independent. One interesting area of future work is to evaluate the performance of the denoising techniques over other types of imaging modalities, such as photometric stereo images [42]. This can help to find suitable denoising methods for use with face datasets employing

various imaging modalities. Although this paper has focused on datasets with few training samples, other possible research would be to explore the noise-robustness of other facial recognition algorithms, such as the deep learning-based and sparse classifiers, which the latter relies on higher number of per subject samples. While the results provided in this paper were mostly based on identification performance, it would be interesting to explore the verification scenarios performance, when different noisy samples are utilised.

REFERENCES

[1] D. Colbry and G. Stockman, "Real-time identification using a canonical face depth map," *IET Computer Vision*, vol. 3, no. 2, pp. 74–92, 2009.

[2] A. Flint, A. Dick, and A. van den Hengel, "Local 3D structure recognition in range images," *IET Computer Vision*, vol. 2, no. 4, pp. 208–217, 2008.

[3] M. Emambakhsh, J. Gao, and A. Evans, "An evaluation of denoising algorithms for 3D face recognition," in *Proceedings of the 5th IET International Conference on Imaging for Crime Detection and Prevention (ICDP 2013)*, pp. 1–6, 2013.

[4] B. Efraty, E. Bilgazyev, S. Shah, and I. A. Kakadiaris, "Profile-based 3D-aided face recognition," *Pattern Recognition*, vol. 45, no. 1, pp. 43–53, 2012.

[5] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, L. Yunliang, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 640–649, 2007.

[6] G. Passalis, P. Perakis, T. Theoharis, and I. Kakadiaris, "Using facial symmetry to handle pose variations in real-world 3D face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1938–1951, 2011.

[7] P. Perakis, G. Passalis, T. Theoharis, and I. Kakadiaris, "3D facial landmark detection under large yaw and expression variations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1552–1564, 2013.

[8] Y. Lei, M. Bennamoun, M. Hayat, and Y. Guo, "An efficient 3D face recognition approach using local geometrical signatures," *Pattern Recognition*, vol. 47, no. 2, pp. 509–524, 2013.

[9] A. Mian, M. Bennamoun, and R. Owens, "An efficient multimodal 2D-3D hybrid approach to automatic face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1927–1943, 2007.

[10] K. I. Chang, K. Bowyer, and P. Flynn, "Adaptive rigid multi-region selection for handling expression variation in 3D face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 157, 2005.

[11] X. Li and F. Da, "Efficient 3D face recognition handling facial expression and hair occlusion," *Image and Vision Computing*, vol. 30, no. 9, pp. 668–679, 2012.

[12] Y. Ming and Q. Ruan, "Robust sparse bounding sphere for 3D face recognition," *Image and Vision Computing*, vol. 30, no. 8, pp. 524–534, 2012.

[13] Y. Wang, J. Liu, and X. Tang, "Robust 3D face recognition by local shape difference boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1858–1870, 2010.

[14] C. Xu, S. Li, T. Tan, and L. Quan, "Automatic 3D face recognition from depth and intensity Gabor features," *Pattern Recognition*, vol. 42, no. 9, pp. 1895–1905, 2009.

[15] M. Emambakhsh, A. Evans, and M. Smith, "Using nasal curves matching for expression robust 3D nose recognition," in *Proceedings of the 6th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, 2013.

[16] H. Mohammadzade and D. Hatzinakos, "Iterative closest normal point for 3D face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 381–397, 2013.

[17] M. Emambakhsh and A. Evans, "Self-dependent 3D face rotational alignment using the nose region," in *Proceedings of the 4th IET International Conference on Imaging for Crime Detection and Prevention (ICDP)*, pp. 1–6, 2011.

[18] A. Moorhouse, A. Evans, G. Atkinson, J. Sun, and M. Smith, "The nose on your face may not be so plain: Using the nose as a biometric," in *3rd IET International Conference on Crime Detection and Prevention (ICDP)*, pp. 1–6, 2009.

[19] Y. Lei, M., and A. A. El-Sallam, "An efficient 3D face recognition approach based on the fusion of novel local low-level features," *Pattern Recognition*, vol. 46, no. 1, pp. 24–37, 2013.

[20] H. Drira, B. B. Amor, A. Srivastava, M. Daoudi, and R. Slama, "3D face recognition under expressions, occlusions, and pose variations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2270–2283, 2013.

[21] R. Llonch, E. Kokiopoulou, I. Tosic, and P. Frossard, "3D face recognition using sparse spherical representations," in *19th IEEE International Conference on Pattern Recognition*, pp. 1–4, 2008.

[22] D. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[23] J. S. Lim, *Two-dimensional Signal and Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990. p. 548.

[24] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[25] M. Emambakhsh, H. Ebrahimnezhad, and M. Sedaaghi, "Integrated region-based segmentation using color components

and texture features with prior shape knowledge," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 4, pp. 711–726, 2010.

[26] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Noise in 3D laser range scanner data," in *IEEE International Conference on Shape Modeling and Applications*, pp. 37–45, 2008.

[27] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[28] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[29] T. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, vol. 1857 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin Heidelberg, 2000.

[30] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems 14*, pp. 841–848, MIT Press, 2002.

[31] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[32] A. Colombo, C. Cusano, and R. Schettini, "UMB-DB: A database of partially occluded 3D faces," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2113–2119, 2011.

[33] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," in *Biometrics and Identity Management*, vol. 5372, pp. 47–56, Springer Berlin / Heidelberg, 2008.

[34] V. Struc, "The PhD face recognition toolbox," February 2012. http://www.mathworks.co.uk/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox, access time: 13-11-2014.

[35] F. D'Almeida, "Nonlinear diffusion toolbox," July 2003. http://www.mathworks.co.uk/matlabcentral/fileexchange/3710-nonlinear-diffusion-toolbox, access time: 19-Oct-2014.

[36] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[37] W. Deng, J. Hu, and J. Guo, "Extended SRC: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[38] Q. Ke and T. Kanade, "Robust L1-norm factorization in the presence of outliers and missing data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 592599, 2005.

[39] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, pp. 78–86, 2004.

[40] S. Berretti, A. Del Bimbo, and P. Pala, "Sparse matching of salient facial curves for recognition of 3-D faces with missing

parts," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 374–389, 2013.

[41] H. Li, D. Huang, J.-M. Morvan, L. Chen, and Y. Wang, "Expression-robust 3D face recognition via weighted sparse representation of multi-scale and multi-component local normal patterns," *Neurocomputing*, vol. 133, no. 0, pp. 179 – 193, 2014.

[42] S. Zafeiriou, M. Hansen, G. Atkinson, V. Argyriou, M. Petrou, M. Smith, and L. Smith, "The Photoface database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 132–139, 2011.